### 题目1：

#include <iostream> // 预处理命令  
using namespace std; // 使用标准命名空间std  
  
// 数组类  
class Array  
{  
private:  
// 数据成员:  
 int \*elem; // 数组元素存储空间  
 int size; // 数组元素个数  
public:  
// 公有函数:  
 Array(int a[], int sz): elem(a), size(sz){ } // 构造函数  
 int GetSize(){ return size; } // 返回数组元素个数  
 int &operator[](int pos){ return elem[pos - 1]; } // 重载下标运算符[]  
};  
  
int main(void) // 主函数main(void)  
{  
 int a[] = {1, 2, 3, 4, 5}; // 定义数组a  
 Array obj(a, 5); // 定义数组对象  
  
 obj[1] = 8; // 为数组元素赋值  
 for (int i = 1; i <= obj.GetSize(); i++) // 依次输出数组各元素之值  
 cout << obj[i] << " "; // 输出第i个元素  
 cout << endl; // 换行  
  
 system("PAUSE"); // 调用库函数system( )，输出系统提示信息  
 return 0; // 返回值0, 返回操作系统  
}

### 题目二：

#include <iostream> // 预处理命令  
using namespace std; // 使用标准命名空间std  
  
// 复数类  
class Complex  
{  
private:  
// 数据成员:  
 double realPart; // 实部  
 double imagePart; // 虚部  
  
public:  
// 公有函数:  
 Complex(double real = 0, double image = 0): realPart(real), imagePart(image){ } // 构造函数  
 double GetRealPart() const{ return realPart; } // 返回实部  
 double GetImagePart() const{ return imagePart; } // 返回虚部  
 \_\_[1]\_\_ operator+(const Complex &a) const // 重载加法运算符+  
 {  
 Complex b; // 定义复数对象  
 b.realPart = this->realPart + a.realPart; // 和的实部  
 b.imagePart = this->imagePart + a.imagePart; // 和的虚部  
 return \_\_[2]\_\_ ; // 返回和  
 }  
};  
  
int main(void) // 主函数main(void)  
{  
 Complex a(1, 2), b(2, 6), c; // 定义复数对象  
  
 c = a + b; // 复数加法运算  
 cout << "a=" << a.GetRealPart() << "+" << a.GetImagePart() << "i" << endl; // 显示a  
 cout << "b=" << b.GetRealPart() << "+" << b.GetImagePart() << "i" << endl; // 显示b  
 cout << "c=" << c.GetRealPart() << "+" << c.GetImagePart() << "i" << endl; // 显示c  
  
 system("PAUSE"); // 调用库函数system( )，输出系统提示信息  
 return 0; // 返回值0, 返回操作系统  
}

### 题目三：

#include <iostream> // 预处理命令 //2  
using namespace std; // 使用标准命名空间std //3  
 //4  
// 整型类 //5  
class Integer //6  
{ //7  
private: //8  
// 数据成员: //9  
 int val; // 整数值 //10  
 //11  
public: //12  
// 公有函数: //13  
 Integer(){ val = 0; } // 无参数的构造函数 //14  
 Integer(int v = 0){ val = v; } // 带参数的构造函数 //15  
 void Show(){ cout << val << endl; } // 显示整数值 //16  
}; //17  
 //18  
int main(void) // 主函数main(void) //19  
{ //20  
 Integer a(2), b; // 定义整型对象 //21  
 a.Show(); // 显示a //22  
 b.Show(); // 显示b //23  
 //24  
 system("PAUSE"); // 调用库函数system( )，输出系统提示信息 //25  
 return 0; // 返回值0, 返回操作系统 //26  
}

### 作业一

设计一个日期类Date, 要求：

（1）包含年(year)、月(month)和日(day)私有数据成员。

（2）包含构造函数，重载关于一日期加上天数的加法运算符+、重载关于一日期减去天数的减加运算符-、重载输出运算符<<与输入运算符>>等。

### 作业二

设计一个时间类Time，要求：

（1）包含时(hour)、分(minute)和秒(second)私有数据成员。

（2）包含构造函数，重载关于一时间加上另一时间的加法运算符+、重载关于一时间减去另一时间的减加运算符-、重载输出运算符<<与输入运算符>>等。